SPERRY RAND

UNIVAC

9200/9200 II
9300/9300 II
SYSTEMS

GENERALIZED COMMUNICATIONS CONTROL ROUTINE

PROGRAMMERS
REFERENCE

UP-7816

This manual is published by the Univac Division of Sperry Rand Corporation in loose leaf format. This format provides a rapid and complete means of keeping recipients apprised of UNIVAC Systems developments. The information presented herein may not reflect the current status of the product. For the current status of the product, contact your local Univac Representative.

The Univac Division will issue updating packages, utilizing primarily a page-for-page or unit replacement technique. Such issuance will provide notification of hardware or software changes and refinements. The Univac Division reserves the right to make such additions, corrections, and/or deletions as, in the judgment of the Univac Division, are required by the development of its Systems.

CONTENTS

CONTENTS 1 to 4

1. INTRODUCTION 1-1 to 1-4

1.1. GENERAL 1-1

1.2. MACRO INSTRUCTIONS 1-1
1.2.1. Declarative Macro Instructions 1-2
1.2.2. Imperative Macro Instructions 1-3

1.3. PROGRAMMING CONVENTIONS 1-3

1.4. STATEMENT CONVENTIONS 1-4

2. GENERALIZED COMMUNICATIONS CONTROL ROUTINE 2-1 to 2-13

2.1. GENERAL 2-1

2.2. COMPATIBILITY 2-1

2.3. MINIMUM HARDWARE AND SOFTWARE CONFIGURATIONS 2-1

2.4. STORAGE REQUIREMENTS 2-3

2.5. USER PROGRAM INTERFACE REQUIREMENTS 2-5

2.6. OPERATOR INTERFACE REQUIREMENTS 2-5

2.7. DECLARATIVE MACRO INSTRUCTION 2-5

2.8. IMPERATIVE MACRO INSTRUCTIONS 2-11
2.8.1. OPEN Macro Instruction 2-11
2.8.2. GET Macro Instruction 2-12
2.8.3. PUT Macro Instruction 2-12
2.8.4. CLOSE Macro Instruction 2-13

3. GCCR FILE CONTROL TABLE AND I/O FUNCTION PACKET FORMAT 3–1 to 3–21

3.1. GENERAL 3–1

3.2. GCCR FILE CONTROL TABLE 3–1
3.2.1. Entry Relay Instruction Area 3–3
3.2.2. Input File Control Area 3–3
3.2.2.1. Input Buffer Address (Bytes 28–29) 3–3
3.2.2.2. Current Input Area Address (Bytes 30–31) 3–3
3.2.2.3. Input Time Countdown (Bytes 32–33) 3–3
3.2.2.4. Input Message Length Countdown (Bytes 34–35) 3–4
3.2.2.5. Buffer Control Word Address (Bytes 36–37) 3–4
3.2.2.6. Input Interrupt Pack Address (Bytes 38–39) 3–4
3.2.2.7. Input Indicator Code Address (Bytes 40–41) 3–4
3.2.2.8. Input Unit Activity Value (Bytes 42–43) 3–4
3.2.2.9. First Packet Address (Bytes 44–45) 3–4
3.2.2.10. End Chain Packet Address (Bytes 46–47) 3–4
3.2.2.11. Control (Byte 48) 3–4
3.2.2.12. Receive Channel (Byte 49) 3–5
3.2.2.13. Latest Packet Address (Bytes 50–51) 3–5
3.2.2.14. Indicator Code Packet Address (Bytes 52–53) 3–5
3.2.2.15. Error Sense Bytes (Bytes 54–55) 3–6
3.2.3. Output File Control Area 3–6
3.2.4. Control Area 3–6
3.2.4.1. Current Input Buffer Address (Bytes 84–85) 3–6
3.2.4.2. Input Move Length (Byte 86) 3–6
3.2.4.3. Output Move Length (Byte 87) 3–6
3.2.4.4. Current Output Buffer Address (Bytes 88–89) 3–8
3.2.4.5. RDEVA (Byte 90) 3–8
3.2.4.6. TDEVA (Byte 91) 3–8
3.2.4.7. Allocation Code (Bytes 92–93) 3–8

3.3. I/O FUNCTION PACKETS 3–8
3.3.1. Time and I/O Field 3–9
3.3.2. Function Field 3–9
3.3.3. Status Field 3–13
3.3.4. Address Field 3–13
3.3.5. Message Length Field 3–16
3.3.6. User Control Field 3–16
3.3.7. Control Field 3–16
3.3.8. Chain Field 3–17
3.3.9. Typical Function Packet Coding 3–17

3.4. MULTIPLE REQUEST PACKET 3–19

3.5. INPUT AND OUTPUT DATA TRANSFER 3–20
3.5.1. Input Data Transfer 3–20
3.5.2. Output Data Transfer 3–20

4. PACKET ADVANCE 4–1 to 4–4

4.1. GENERAL 4–1

4.2. PACKET ADVANCE (EXCLUDING PACKET CHAINING) 4–1

4.3. PACKET ADVANCE (INVOLVING PACKET CHAINING) 4–2
4.3.1. Packet Chaining in I/O Function Packets 4–2
4.3.2. Packet Chaining in Multiple Request Packets 4–2
4.3.2.1. Half-Duplex Mode Multiple Request Considerations 4–3
4.3.2.2. Full-Duplex Mode Multiple Request Considerations 4–4

4.4. THE EFFECT OF CONTROL FUNCTIONS ON PACKET ADVANCE 4–4

4.5. THE EFFECT OF INDICATOR CODINGS ON PACKET ADVANCE 4–4

5. ERROR CONDITIONS AND RECOVERY 5–1 to 5–5

5.1. GENERAL 5–1

5.2. ERROR RECOVERY INVOLVING OPERATOR INTERVENTION 5–1

5.3. USE OF INDICATOR CODING FOR ERROR RECOVERY 5–2
5.3.1. Use of I/O Register 5–2
5.3.2. Options Available to the Problem Program 5–3
5.3.3. Entry and Exit of Indicator Coding 5–3
5.3.4. GCCR Action Resulting from Indicator Code Status 5–4

6. PROGRAMMING CONSIDERATIONS 6–1 to 6–3

6.1. GENERAL 6–1

6.2. INITIALIZING A MANUAL DIAL RECEIVE PROGRAM 6–1

6.3. INITIALIZING A MANUAL DIAL, HALF-DUPLEX TRANSMIT PROGRAM 6–2

6.4. AUTOMATIC DIALING 6–2

6.5. CONTINUOUS READ PAPER TAPE ASRS 6–3

6.6. AUTOMATIC ANSWERING RESTRICTION 6–3

FIGURES

3–1. GCCR File Control Table Format 3–2

3–2. I/O Function Packet Format 3–8

3–3. Initial Command Selection Sequence 3–14

3–4. Typical Output Function Packet Coding 3–18

3–5. Typical Input Function Packet Coding 3–18

3–6. Multiple Request Packet Format 3–19

4–1. Packet Advance Sequence for Multiple Request 4–3

TABLES

2-1. GCCR – Compatible Remote Terminals and DCS Features Permitted with Each 2-2

2-2. Storage Requirements for Communication Programs Using GCCR 2-4

2-3. Summary of DTFGC Macro Instruction Keyword Parameters 2-10

3-1. First Error Sense Byte Repertoire 3-7

3-2. Second Error Sense Byte Repertoire 3-7

3-3. Command Byte Repertoire 3-10

3-4. Status Byte Coding for I/O Function Packet 3-16

3-5. Control Byte Coding for I/O Function Packet 3-17

5-1. Recoverable Display Stops 5-2

5-2. Nonrecoverable Display Stops 5-2

1. INTRODUCTION

1.1. GENERAL

This manual describes the Generalized Communications Control Routine (GCCR) provided for a UNIVAC 9200/9300 System/Data Communications Subsystem (DCS) user. The manual includes descriptions concerning hardware/software requirements and configurations, routine compatibility, storage requirements, user and operator interfaces, and the macro instructions necessary to implement the GCCR in a particular program.

A knowledge of the *UNIVAC 9200/9200 II/9300/9300 II Systems Card Assembler Programmers Reference, UP-4092* (current version), *UNIVAC 9200/9200 II/9300/9300 II Systems Minimum Operating System Programmers Reference, UP-4547* (current version), and the *UNIVAC 9200/9200 II/9300/9300 II System Card System IOCS Programmers Reference, UP-7728* (current version) is helpful in using the manual.

The manual has six sections. A discussion of each section's contents is as follows:

- Section 1 contains introductory information required for understanding the format of the macro instructions, the program conventions, and the statement conventions used within the manual.
- Section 2 contains information pertaining to the use, requirements, and implementation of the GCCR from the user's point of view.
- Section 3 contains information about the structure and purpose of the tables and packets comprising the GCCR, and descriptions of the events which take place during data input and output transfers.
- Section 4 contains information pertaining to the manner in which the GCCR handles packet advance for function packets submitted through GET and PUT imperative macro instructions and through multiple request packets. Information concerning packet chaining and the effects of control function execution upon packet advance are also discussed.
- Section 5 contains information pertaining to error indications and recovery. Operator response as well as program response to error conditions are also discussed.
- Section 6 contains programming considerations which should be taken into account when communication is established between terminals and a UNIVAC 9200/9300 central processor site.

1.2. MACRO INSTRUCTIONS

A macro instruction is similar in form to a source code instruction. It may or may not have a label, but it must have an operation code and an operand field containing one or more parameters.

The parameters used with the declarative macro instructions describe all aspects of the file to be processed; the parameters used with the imperative macro instructions point to the file described by a declarative macro and sometimes add additional details specifying processing action to be taken.

1.2.1. Declarative Macro Instructions

A problem program informs the system of the parameters, special conditions, current status, and options pertaining to a file. This is accomplished by including a declarative (file definition) macro instruction for each file required by the problem program. These macro instructions generate nonexecutable code, such as constants and storage areas for variables. Therefore, these macro instructions should be separated physically from the inline file processing coding. The declarative macro instruction and the selected keyword parameters in the operand define the file. The first three characters of the operation code are DTF, meaning Define The File. The last two characters indicate the type of device or method of accessing. A keyword parameter consists of a word or code immediately followed by an equals (=) sign which is, in turn, followed by one specification.

The format of the declarative macro instruction is:

LABEL	ƀ OPERATION ƀ	OPERAND
filename	DTFxx	keyword–1 = x,keyword–2 = y,...,keyword–n = z

The symbolic name of the file must appear in the label field. It has a maximum of four characters and must begin with an alphabetic character. The appropriate DTF designation must appear in the operation field. The keyword parameters are written in any order in the operand field and must be separated by commas. Appropriate assembler rules regarding macro instructions apply to blank columns and continuation statements.

The alternate form of writing the declarative macro instructions is:

LABEL	ƀ OPERATION ƀ	OPERAND	72
filename	DTFxx	keyword–1 = x,	X
		keyword–2 = y,	X
		.	.
		.	.
		.	.
		keyword–n = z	

In the alternate form, a continuation mark is necessary in column 72 of every line except the last. Each keyword parameter and specification, except the last, must be followed by a comma.

1.2.2. Imperative Macro Instructions

A problem program must communicate with IOCS in order to accomplish the processing of files that have been defined by declarative macro instructions. This is accomplished by including imperative (file processing) macro instructions in the problem program, which in turn communicate with the IOCS. The imperative macro instructions are expanded as inline executable code. Not all macro instructions are available for use on all devices. Some are specifically input type macro instructions and cannot be used for a device that is exclusively used for output; the opposite is true for output type instructions. See the detailed descriptions of particular IOCS for the use of imperative macro instructions.

The format of the imperative macro instructions is:

LABEL	ƀ OPERATION ƀ	OPERAND
[name]	xxxx	yyyy,...,zzzz

A symbolic name can appear in the label field. It can have a maximum of four characters and must begin with an alphabetic character. The appropriate verb or code must appear in the operation field. The positional parameters (as signified by the name) must be written in the specified order in the operand field and be separated by commas. When a positional parameter is omitted, the comma must be retained to indicate the omission except in the case of omitted trailing parameters. Appropriate assembler rules regarding macro instructions apply to blank columns and continuation statements.

1.3. PROGRAMMING CONVENTIONS

A user routine may be required in the main source program that is accessed by the IOCS when certain checking features are required (for example, printer overflow). IOCS automatically stores the program reentry address in register 14 when the branch to the user routine occurs. The user routine is therefore required to provide the necessary return linkage to the main source program. If the user routine utilizes register 14, it must preserve and restore register 14 before terminating. This also must be done if any macro instruction is executed by the user routine, since all macros use program registers 14 and 15. If register 14 is not preserved, the reentry address is lost. Register 15 also may be used by the user routine and it need not be preserved. However, its contents are altered by the execution of any macro instruction.

1.4. STATEMENT CONVENTIONS

The conventions used to illustrate statments in the manual are as follows:

- Capital letters and punctuation marks (except braces, brackets, and ellipses) are information that must be coded exactly as shown.
- Lowercase letters and terms represent information that must be supplied by the programmer.
- Information contained within braces represents necessary entries, one of which must be chosen.
- Information contained within brackets represents optional entries that (depending on program requirements) are included or omitted. Braces within brackets signify that one of the entries must be chosen if that operand is included.
- An ellipsis indicates the presence of a variable number of entries.
- In the coding of macros, commas are required after each parameter except after the last parameter specified. When a positional parameter is omitted from within a series of parameters, the comma must be retained to indicate the omission.

2. GENERALIZED COMMUNICATIONS CONTROL ROUTINE

2.1. GENERAL

The UNIVAC Generalized Communications Control Routine (GCCR) is a logical input/output control routine designed to function as the software interface between a user's problem program and the various remote communication devices of the DCS to which his system is connected. The GCCR enables a user to request the execution of input/output operations without assuming the responsibilities of handling the hardware. The user can, by the use of declarative and imperative macro instructions, develop a communications handler tailored to meet the needs of his particular system configuration and problem program requirements.

2.2. COMPATIBILITY

The GCCR can be used with all configurations of the UNIVAC 9200/9300 Systems. This flexibly designed communications subroutine is capable of controlling up to eight communication lines and can be used with any of the terminals connected with either the DCS-1 or DCS-4 configuration.

Another feature making the GCCR compatible with all system configurations is its ability to communicate in both half-duplex and full-duplex modes at telegraph, voice-grade, and wideband transmission speeds with devices, such as keyboard printers, scopes, teletypewriters, and other processors. The capability of being used in both the symbiont and the main chain modes permits the GCCR to be involved in background batch processing as well as in the more communication-active inquiry and response types of subroutines.

The availability of current UNIVAC 9200/9300 IOCS subroutines makes it possible to interface the GCCR with I/O devices to perform operations, such as remote-to-printer, disc-to-remote, and remote-to-tape.

2.3. MINIMUM HARDWARE AND SOFTWARE CONFIGURATIONS

The minimum hardware configuration required for utilization of the GCCR is a UNIVAC 9200/9300 central processor unit with a main storage capacity of at least 8K, a card reader, a DCS-1 or DCS-4, and any peripheral device required by the user problem program. Note that the GCCR element requires 16K of main storage to perform a preassembly macro pass or an assembly. Once assembled and linked to the DTFGC instruction, the GCCR and applicable subroutines can be operated in an 8K main storage environment. The configuration of the DCS-1 and DCS-4 is comprised of any of the features listed in Table 2-1. (Table 2-1 lists all compatible remote devices available with their respective line terminal.) The large scale systems (UNIVAC 418, 494, 1107, and 1108) are not listed in Table 2-1, because the GCCR is not designed to replace the REM-1 library nor to serve the needs of the advanced user who wishes to write his own remote terminal program for a UNIVAC 9000 System.

	LINE TERMINAL																			
	LTC		LRC		DA	LT										CI				ATA
REMOTE TERMINAL	F-1000-00 for LTC-1	8575-00 for LTC-4	F1008-99 for LTC-1	F1008-00 for LTC-4	F1007-99	F1003-99 LT TELEGRAPH NONCHECKING	F1003-98 LT TELEGRAPH CHECKING	F1003-97 LT TELEX	F1003-96 LT TWX	F1004-99 LT MEDIUM SPEED NONCHECKING	F1004-98 LT MEDIUM SPEED CHECKING	F1005-99 LT SYNCHRONOUS NONCHECKING	F1005-98 LT SYNCHRONOUS CHECKING	F1005-97 LT REMOTE COMPUTER	F1006-99 LT PARALLEL	F1002-00 CI TELEGRAPH	F1002-03 CI PRIVATE LINE	F1002-04 CI DATA-PHONE +	F1002-05 CI WIDEBAND	F1010-99 ASYNCHRONOUS TIMING ASSEMBLY
UNIVAC 1004/DLT-1	■	■	■	■								■	■				■	■		
UNIVAC 1004/DLT-1B	■	■	■	■								■	■						■	
UNIVAC 1004/DLT-2	■	■	■	■								■	■				■	■		
UNIVAC 1004/DLT-3	■	■	■	■								■	■				■	■		
UNIVAC 1005/DLT-1	■	■	■	■								■	■				■	■		
UNISCOPE 100/DCT 1000	■	■	■	■	■	■	■			■	■	■	■				■	■		■
UNISCOPE 300	■	■	■	■								■	■				■	■		
UNIVAC DCT 500	■	■			■	■	■			■	■						■	■		■
UNIVAC DCT 2000	■	■	■	■	■							■	■				■	■		
All UNIVAC 9000 Series Systems	■	■	■	■	■							■	■	■			■	■	■	
TELETYPE + Models 28, 32, 33, 35, 37	■	■			■	■	■									■	■	■		■
TWX +	■	■			■				■											■
TELEX +	■	■						■												■
804C and 804K TOUCH-TONE + Telephones	■	■													■					

+TELETYPE – Trademark of Teletype Corporation
TELEX – Trademark of Western Union Telegraph Co.
TOUCH-TONE – Service Mark of American Telephone and Telegraph
TWX – Trademark of American Telephone and Telegraph Co.
DATA PHONE – Trademark of American Telephone and Telegraph Co.

Table 2–1. GCCR – Compatible Remote Terminals and DCS Features Permitted With Each

The minimum software configuration required for using the GCCR is the Minimum Operating System (MOS). The GCCR is also designed to function in both the Non-current Operating System (NCOS) and the Concurrent Operating System (COS). IOCS routines available for use with the GCCR are:

- Drum Printer
- Integrated Printer
- Integrated Reader
- On-Line UNIVAC 1001 Controller
- Serial Punch
- Row Punch
- Paper Tape Reader
- Paper Tape Punch
- Magnetic Tape
- UNIVAC 8410 Disc
- UNIVAC 8411 Disc
- UNIVAC 8414 Disc

2.4. STORAGE REQUIREMENTS

The amount of storage required for a communication program handled by the GCCR is based upon the type, configuration, and complexity of the system being used. A factor to be considered when determining storage requirements is the number of remote terminals employed. The amount of buffer required for each remote terminal must be sufficient to contain a portion of message considered excessively long or to contain the entire message for those considered relatively short. User coding must also be considered. For example, user coding is required to test and reset the indications provided in the status bytes of the I/O function packets used by the GCCR; the user coding is also required to initiate activities, such as processing a subsequent record, initiating a retransmission of data, or cancelling a program. The activity initiated by the user program must be appropriate to the status detected. When IOCS routines are taken into account, storage requirements vary with respect to the size of the specific IOCS routines used.

A list of the storage requirements for a communication program using the GCCR is provided in Table 2-2.

STORAGE REQUIREMENT (bytes)	USE
2400	GCCR
376	For each remote terminal (KSR, ASR, RO, UNISCOPE, and so forth); consists of the following: – Buffers: Two 128-byte buffers for full-duplex operation, or for half-duplex operation when optionally separate input and output buffers are utilized. One 128-byte buffer for half-duplex operation (used for both input and output). – Work area: A work area equal to the largest message expected during transfer of data is required. When operating in a full-duplex system, two areas must be specified: one for largest input message and one for largest output message. – I/O function packets: 12 bytes for each unique I/O function required.
100 (approximate)	Optional user coding for DCS interrupt response
1600	MOS
4000	NCOS
5000	COS
	IOCS Subroutine
600	Drum Printer
400	Integrated Printer
720	On-Line UNIVAC 1001 Controller
550	Integrated Reader
675	Serial Punch
950	Row Punch
600 (approximate)	Paper Tape Reader
600 (approximate)	Paper Tape Punch
1200-3100	Magnetic Tape
1300-5800	UNIVAC 8410 Disc
1600-7200	UNIVAC 8411/8414 Disc

Table 2–2. Storage Requirements for Communication Program Using GCCR

2.5. USER PROGRAM INTERFACE REQUIREMENTS

Interface between the GCCR and the user problem program consists of two requirements, establishing a file description to the IOCS for each file the user program must access and providing the instructions which direct the sequential processing and operation of the GCCR.

The user establishes a file description to the IOCS programs by means of the DTFGC declarative macro instruction. The DTFGC macro instruction produces, through the preassembly macro pass, a file control table (see Section 3) for the file described by the keyword parameters provided within the DTFGC macro instruction. The output of the preassembly macro pass may be assembled with the problem program or assembled alone and linked with the problem program. The user, however, must provide a DTFGC declarative macro instruction for each file to be accessed by the user's problem program.

The operation of the GCCR is directed by means of imperative macro instructions which are specified within the user's problem program. The imperative macro instructions are executed to accomplish specific I/O functions in proper relationship to the requirements of the problem program. The use of the imperative macro instructions differs from their conventional use in that the imperative macro instructions identify function packets rather than I/O areas. The function packet identified contains a detailed description of the specific I/O order to be executed as well as the control information necessary to execute that order.

2.6. OPERATOR INTERFACE REQUIREMENTS

The interface between the operator and the GCCR is primarily the responsibility of the problem program. The GCCR, however, provides the problem program with the status of the functions executed; the problem program must provide for the interpretation of this status information in determining if operator intervention is required.

In some instances, the operator is required to keyin a response or to react to a specific error halt display which signifies that a hardware or software error has been encountered.

The user may provide for the handling of error conditions by the use of indicator coding included in the problem program. The purpose of indicator coding is to inform the problem program of changes in the status of I/O functions and to direct the initiation of a predetermined error recovery procedure to remedy the situation. The use of the user indicator coding is discussed in detail in Section 3.

2.7. DECLARATIVE MACRO INSTRUCTION

In order to use the GCCR, the programmer defines to the IOCS each input and output file requiring the use of a communications device. This is accomplished by means of a DTFGC declarative macro instruction. (DTFGC is the mnemonic operation code for *D*efine *T*he *F*ile for the Generalized Communications Control Routine.)

The DTFGC macro instruction provides the keyword parameters to which the preassembly macro pass can generate a source code module comprised of a file control table. The keyword parameters of the macro instruction are converted and inserted in the file control table as entries which interface the file to the control routine and which are used to direct the operation of the GCCR. The file control table module produced by the preassembly macro pass may be assembled as part of the user program or may be assembled alone and then linked with the problem program.

The filename entered into the label field of a DTFGC macro instruction must correspond to the symbolic name assigned to the file by the programmer. Each symbolic name assigned is restricted to four characters in length and must conform to the assembler language rules for labels.

The format of the DTFGC macro instruction is shown below with each required or optional keyword parameter in alphabetical order. A description of the parameters and their specifications follows the format, and a summary of the keyword parameters is given in Table 2.3.

The format of the DTFGC macro instruction is:

LABEL	␢ OPERATION ␢	OPERAND
filename	DTFGC	CBUF=label,
		SYMB=n
		[,RACT=nn]
		[,RCHN=nn]
		[,RDEV=nn]
		[,RIND=label]
		[,TACT=nn]
		[,TBUF=label]
		[,TCHN=nn]
		[,TDEV=nn]
		[,TIND=label]

- Communications Buffer

  This keyword parameter is required for input files that are to be executed by the GCCR. It is used to identify a problem program area (128 bytes in length, modulo 128) to be used as a dynamic buffer for receiving input messages and subsequently to transfer this data to the areas specified in the address fields of function packets. The buffer area specified by the communications buffer keyword parameter serves as both an input and output area whenever the TBUF parameter is omitted from the DTFGC macro instruction. The format of the communications buffer keyword parameter is:

  CBUF=label

  where label is the address of the dynamic buffer area.

  NOTE: To ensure that it conforms to the modulo 128 restrictions, the CBUF area is defined in the problem program as follows:

```
        ORG *,128

label DS    CL128
```

■ Output Buffer

This keyword parameter is used to identify a program area (128 bytes in length, modulo 128) to be used as a dynamic buffer for outputting data. The data outputted is transferred from the area specified in the address field of the function packet to a remote terminal. This parameter must be specified for full-duplex modes of operation and for half-duplex modes of operation when separate input and output buffers are employed. The output buffer parameter is to be omitted for simplex modes of operation and half-duplex modes of operation where only one 128-byte buffer is utilized. When only one buffer is employed, the area specified by the CBUF keyword parameter serves as both the input and output dynamic buffer. The GCCR assumes that a full-duplex environment is intended whenever separate buffers (CBUF and TBUF) are specified.

When operating with separate input and output buffers in a half-duplex environment, caution must be exercised so that commands to both input and output line terminals do not overlap and cause a ''loop back'' situation of transferring data from the output buffer to the input buffer. This condition will cause unrecoverable error situations.

The format of the output buffer keyword parameter is:

```
TBUF=label
```

where label is the address of the dynamic buffer area.

NOTE: To ensure that it conforms to the modulo 128 restrictions, the TBUF area is defined in the problem program as follows:

```
      ORG  *,128
label DS   CL128
```

■ Symbiont

This required keyword parameter is used to specify the allocation code assigned to the communication devices employed by the file. The value specified for this parameter must correspond to the level of the routine with which the communication device is to operate. The format for the symbiont keyword parameter is:

```
SYMB=n
```

where n is a decimal number from 1 to 5 representing the symbiont number when the communications control routine is incorporated into a symbiont, or n is equal to the decimal value of 6 representing the allocation code for the main program.

■ Receive Available Cycle Time

This keyword parameter is required whenever the GCCR handles input messages. It is used to specify the percentage of available memory cycles (activity sum) required by the slowest device in the input communication linkage. Items which must be considered in determining the degree of slowness of a device are the communications channel itself, the line terminal speed, and the modem speed. The format of receive available cycle time keyword parameter is:

```
RACT=nn
```

where nn is a value from 01 to 100 percent.

The following table provides typical examples of the values specified for the RACT parameter when associated with data sets having the baud rates listed.

DATA SET	BAUD RATE	VALUE OF RACT (Percent)
201A3	2000	1
201B3 or 202D	2400	1
205B1	4800	1
301B or 303C	50,000	10

- Receive Channel Entry

This keyword parameter is used whenever an input message is handled by the GCCR. When specified, the parameter identifies the multiplexer subchannel to which the receiver of the data communications subsystem is connected and must agree with the PUTBL entry of the operating system.

The format for the receive channel entry parameter is:

RCHN=nn

where nn may equal any odd numbered decimal integer between the values of 17 and 31.

NOTE: As previously stated, the value specified for the RCHN parameter must agree with the PUTBL entry of the operating system. For example, RCHN would be specified as RCHN=25 if the following PUTBL was used when generating the logical unit/physical unit tables of the operating system.

```
PUTBL    CDVC,25,0,0,4,B,24
```

- Receive Device

This keyword parameter identifies the logical unit number of the input communications device connected to the multiplexer subchannel specified by the RCHN parameter and must agree with the PUTBL entry of the operating system. The keyword must be specified whenever the RCHN parameter is specified. The format of the receive device parameter is as follows:

RDEV=nn

where nn is expressed as a decimal number or as a two-digit hexadecimal number having an X'nn' format. The value for nn is any number from 00 through 63.

NOTE: As previously stated, the value of RDEV must agree with the PUTBL entry of the operating system. For example, RDEV would be specified as RDEV=4 if the following PUTBL was used when generating the logical unit and physical unit tables of the operating system.

```
PUTBL    CDVC,25, , ,4,B,24
```

- Receive Indicator Code

This keyword parameter is used only when the problem program employs the use of indicator coding to handle interrupts resulting from error conditions; error conditions may occur while receiving an input data transfer from a remote communications device. When specified, this parameter identifies the problem program indicator code routine address. The format of the receive indicator code is:

RIND=label

where label is the tag of the first instruction to be executed in the I/O mode for interrupt conditions resulting from data input from remote communication devices.

- Transmit Available Cycle Time

This keyword parameter is required whenever the GCCR handles output messages. The parameter specifies the percentage of available memory cycles required by the slowest device in the output communication linkage. Items to be considered in determining the degree of slowness of a device are the communications channel itself, the line terminal speed, and the modem speed. The format of transmit available cycle time keyword parameter is:

TACT=nn

where nn is a value from 01 to 100 percent.

The following table provides typical examples of the values specified for the TACT parameter when associated with the data sets having the baud rates listed.

DATA SET	BAUD RATE	VALUE OF TACT (Percent)
201A3	2000	1
201B3 or 202D	2400	1
205B1	4800	1
301B or 303C	50,000	10

- Transmit Channel Entry

This keyword parameter is used whenever an output message is handled by the GCCR. When specified, the parameter identifies the multiplexer subchannel to which the transmitter of the data communications subsystem is connected. The format for the transmit channel entry parameter is:

TCHN=nn

where nn may equal any even numbered decimal integer between the values of 16 and 30.

- Transmit Device

This keyword parameter identifies the logical unit number of the output communications device connected to the multiplexer subchannel specified by the TCHN parameter. When specified, the TCHN parameter must agree with the PUTBL entry of the operating system. The keyword parameter must be specified whenever the TCHN parameter is specified. The format of the transmit device parameter is:

TDEV=nn

where nn may be expressed as a decimal number or as a two-digit hexadecimal number having an X'nn' format. The value for nn is any number from 00 through 63.

NOTE: As previously stated, the value of TDEV must agree with the PUTBL entry of the operating system. For example, TDEV would be specified as TDEV=4 if the following PUTBL was used when generating the logical unit and physical unit tables of the operating system.

```
PUTBL    CDVC,25, , ,4,B,24
```

- Transmit Indicator Code

This keyword parameter is used only when the problem program employs the use of indicator coding to handle interrupts resulting from error condition; error conditions may occur during the transmission of output data to a remote communications device. When specified, this parameter identifies the problem program indicator code routine address. The format of the transmit indicator code is:

TIND=label

where label is the tag of the first instruction to be executed in the I/O mode for interrupt conditions resulting from data output to a remote communications device.

KEYWORD	SPECIFICATION	FILES		REMARKS
		INPUT	OUTPUT	
CBUF	label	R	R	Identifies dynamic buffer area
SYMB	{ 1 to 5 / 6 }	R	R	Identifies GCCR as a symbiont or as part of the main program
RACT	1 to 100	R		Percentage of memory cycles available for slowest input device
RCHN	nn = odd number subchannel (17 to 31)	R		Identifies receiver subchannel
RDEV	nn = logical unit	R		Identifies logical unit number of receiver
RIND	label	X		Specifies use of indicator coding for input interrupt handling
TACT	1 to 100		R	Percentage of memory cycles available for slowest output device
TCHN	nn = even number subchannel (16 to 30)		R	Identifies transmitter subchannel
TDEV	nn = logical unit		R	Identifies logical unit number of transmitter
TIND	label		X	Specifies use of indicator coding for output interrupt handling

R = required
X = optional

Table 2–3. Summary of DTFGC Macro Instruction Keyword Parameters

Example:

LABEL (1)	␢ OPERATION ␢ (10)	OPERAND (16)		72	80
INPU	DTFGC	CBUF=INLT,		X	
		TBUF=OPLT,	OPTIONAL	X	
		SYMB=6,		X	
		RACT=1,		X	
		RCHN=25,		X	
		RDEV=4,		X	
		RIND=INDI,	OPTIONAL	X	
		TACT=1,		X	
		TCHN=24,		X	
		TDEV=4,		X	
		TIND=INDP	OPTIONAL		

2.8. IMPERATIVE MACRO INSTRUCTIONS

Imperative macro instructions supplied for the GCCR are used to open, close, and process the files required by the problem program. By including the appropriate imperative macro instructions within the problem program, the user directs the operation of the GCCR so that specific I/O functions are accomplished in proper relation to the requirements of the problem program.

The imperative macro instructions supplied for the GCCR are OPEN, GET, PUT, and CLOSE. Optional labels for these macro instructions are restricted to a maximum of four characters.

2.8.1. OPEN Macro Instruction

The OPEN imperative macro instruction initializes and prepares the file for processing. This macro must be issued before any other macro instruction pertaining to the same file is issued.

When executed, the OPEN macro instruction initializes the file control table of the named file. No data transfer is initiated however.

The format of the OPEN imperative macro instruction is:

LABEL	␢ OPERATION ␢	OPERAND
[name]	OPEN	filename

- Positional Parameter 1

filename – is the symbolic name of the file to be opened as specified in the label field of the DTFGC declarative macro instruction. The filename may have a maximum of four characters.

Example:

LABEL	OPERATION	OPERAND
	OPEN	CARD
	OPEN	COMM

Initializes the files labeled CARD and COMM.

2.8.2. GET Macro Instruction

The GET imperative macro instruction is used to submit a request for the execution of an input function on the receive channel specified by the RCHN parameter of the DTFGC macro instruction.

The format of the GET imperative macro instruction is:

LABEL	OPERATION	OPERAND
[name]	GET	filename,packet

- Positional Parameter 1

  filename – is the symbolic name of the file to be accessed as defined in the label field of the DTFGC declarative macro instruction.

- Positional Parameter 2

  packet – the symbolic name of the I/O function packet which contains the details for the input function to be executed.

Example:

LABEL	OPERATION	OPERAND
	GET	COMM, INAR

Places a request into the function execution list for the execution of the I/O function packet label INAR for the file labeled COMM.

2.8.3. PUT Macro Instruction

The PUT imperative macro instruction is used to submit a request for the execution of an output function on the transmit channel specified by the TCHN parameter of the DTFGC macro instruction.

The format of the PUT imperative macro instruction is:

LABEL	OPERATION	OPERAND
[name]	PUT	filename,packet

- Positional Parameter 1

  filename – is the symbolic name of the file to be accessed as defined in the label field of the DTFGC declarative macro instruction.

- Positional Parameter 2

  packet – the symbolic name of the I/O function packet which contains the details for the output function to be executed.

Example:

LABEL	OPERATION	OPERAND
	PUT	COMM,OUT

Places a request in the function execution list for the execution of the I/O function packet labeled OUT which transmits to the file labeled COMM.

2.8.4. CLOSE Macro Instruction

The CLOSE imperative macro instruction ensures the proper closing of a file after all processing has been completed. A file may be closed at any time. Once closed, a file can no longer be accessed unless that file is reopened by means of an OPEN macro instruction. To ensure proper closing of a file, the CLOSE macro instruction should be executed before the problem program goes to the end-of-job (EOJ).

The format of the CLOSE imperative macro instruction is:

LABEL	OPERATION	OPERAND
[name]	CLOSE	filename

- Positional Parameter 1

  filename – is the symbolic name of the file to be closed as specified in the label field of the DTFGC declarative macro instruction. The filename may have a maximum of four characters.

Example:

LABEL	OPERATION	OPERAND
	CLOSE	CARD
	CLOSE	COMM

Closes the files labeled CARD and COMM.

3. GCCR FILE CONTROL TABLE AND I/O FUNCTION PACKET FORMATS

3.1. GENERAL

To understand the manner in which the GCCR functions, one must have a basic knowledge of the purpose, structure, and contents of the tables and packets utilized by the GCCR. Descriptions, therefore, are provided for both the GCCR file control table and the I/O function packets comprising the GCCR. In addition, descriptions are provided for the format of the multiple request packet available with the GCCR and for the events occurring during a typical input and output data transfer.

3.2. GCCR FILE CONTROL TABLE

The file control table serves as the primary interface between the user problem program and the GCCR. Through the use of this table, the user defines the characteristics and the amount of control desired for each file serviced by the GCCR. The entries into the file control table are derived from the keyword parameters of the DTFGC macro instruction and from the dynamic operating status recorded by the GCCR. The GCCR uses the contents of the table to issue I/O orders as directed by the user program and to maintain the status of these orders for the problem program.

Each file control table generated has four functional areas: entry relay instruction area, input file control area, output file control area, and control area. Figure 3-1 illustrates the format of the file control table.

Offset			Area
0	BAL 15, J?OP	OPEN	
4	BAL 15, J?CL	CLOSE	
8	BAL 15, J?GT	GET	
12	BAL 15, J?IN	INPUT INTERRUPT	ENTRY RELAY INSTRUCTION AREA
16	BAL 15, J?PT	PUT	
20	BAL 15, J?OT	OUTPUT INTERRUPT	
24	BAL 14, J?QK	CLOCK	
28	INPUT BUFFER ADDRESS	CURRENT INPUT AREA ADDRESS	
32	INPUT TIME COUNTDOWN	INPUT MESSAGE LENGTH COUNTDOWN	
36	BUFFER CONTROL WORD ADDRESS	DC Y(*-26) INPUT INTERRUPT PACK	
40	INPUT INDICATOR CODE ADDRESS	INPUT UNIT ACTIVITY VALUE	INPUT FILE CONTROL AREA
44	FIRST PACKET ADDRESS	END CHAIN PACKET ADDRESS	
48	CONTROL (IP ND PH AS TO OP) / RCHN	ADDRESS LATEST PACKET SUBMITTED	
52	INDICATOR CODE PACKET ADDRESS	ERROR SENSE BYTES	
56	OUTPUT BUFFER ADDRESS	CURRENT OUTPUT AREA ADDRESS	
60	OUTPUT TIME COUNTDOWN	OUTPUT MESSAGE LENGTH COUNTDOWN	
64	BUFFER CONTROL WORD ADDRESS	DC Y(*-46) OUTPUT INTERRUPT PACK	
68	OUTPUT INDICATOR CODE ADDRESS	OUTPUT UNIT ACTIVITY VALUE	OUTPUT FILE CONTROL AREA
72	OUTPUT FIRST PACKET ADDRESS	LATEST OUTPUT PACKET CHAINED	
76	CONTROL (IP ND PH AS TO OP) / TCHN	LATEST OUTPUT PACKET SUBMITTED	
80	INDICATOR CODE PACKET ADDRESS	ERROR SENSE BYTES	
84	CURRENT ADDRESS IN INPUT BUFFER	IN MOVE LENGTH / OUT MOVE LENGTH	
88	CURRENT ADDRESS IN OUTPUT BUFFER	RDEVA / TDEVA	
92	ROUTINE ALLOCATION CODE		WORK AREAS FOR DYNAMIC TRANSFERS AND CONTROL
96			
100			

Figure 3-1. GCCR File Control Table Format

3.2.1. Entry Relay Instruction Area

This functional area comprises the first 28 bytes of the file control table. It contains the seven branch and link (BAL) instructions which interface the GCCR to the imperative macro instructions specified in the user program. The BAL instructions provide the means to access the OPEN, GET, PUT, and CLOSE entrances to the GCCR as well as entrances for input and output interrupt response.

3.2.2. Input File Control Area

The input file control area consists of bytes 28 through 55. The entries made into the fields of this area serve to control the status of the input (receive) channel of the file. In other than full-duplex operation, the packet address fields of this section maintain control of the function packets for input and output messages in order to achieve the proper sequence of execution. The contents and information pertinent to the individual fields of this functional area are described in 3.2.2.1. through 3.2.2.15.

3.2.2.1. Input Buffer Address (Bytes 28-29)

The input buffer address field contains the address of the input buffer as derived from the CBUF parameter of the DTFGC macro. This field is unchanged throughout the operation.

3.2.2.2. Current Input Area Address (Bytes 30-31)

The current input area address field contains the address in the input area to which the next transfer of data will take place from the input buffer. This field is updated after each data transfer by the number of characters transferred. At the completion of an input message or at an error termination, the address specified will be one greater than that of the last character transferred to the input area. The user may access and alter this address in indicator coding so that the content of the buffer is transferred to a location other than the intended input area. However, the destination of all subsequent transfers will be relative to the address in this field.

The length of an input message can be determined by subtracting the input/output area address (bytes 4 and 5) of the function packet from this field.

3.2.2.3. Input Time Countdown (Bytes 32-33)

The input time countdown field contains the binary count of seconds of time allotted to the completion of an input function. The field has three states:

Positive State – A number greater than zero, indicates the remaining time allotted to the execution of the function.

Zero State – Zero time indicates that the time allotted to the function has elapsed.

Negative State – A 1 bit in the most significant bit position indicates that the input function in process is to be unlimited in time or that no input function is currently being timed.

3.2.2.4. Input Message Length Countdown (Bytes 34-35)

This field contains the binary count of the message length remainder. The count is initiated from the function packet and reduced after each data transfer to the input area. If the message length value in the function packet is zero, this field is set to and remains zero. At the successful or error completion of the function, this field will contain the difference between the message length specified in the function packet and the message length actually transferred to the input area.

3.2.2.5. Buffer Control Word Address (Bytes 36-37)

This field contains the address of the input buffer control word. This address is produced in the table by the preassembly macro pass from the RCHN parameter of the DTFGC macro instruction and remains constant throughout processing.

3.2.2.6. Input Interrupt Pack Address (Bytes 38-39)

This field contains the address of the input interrupt entry relay instruction. The field is set by the preassembly macro pass if the parameter RCHN is given and remains constant throughout processing.

NOTE: Bytes 36-39 comprise the Buffer Scan Table Entry of the operating system for the input channel.

3.2.2.7. Input Indicator Code Address (Bytes 40-41)

This field contains the address of the routine submitted to the preassembly macro pass under the RIND parameter. The field remains constant.

3.2.2.8. Input Unit Activity Value (Bytes 42-43)

This field contains the activity value submitted to the preassembly macro pass under the RACT parameter. The field remains constant.

3.2.2.9. First Packet Address (Bytes 44-45)

This field contains the address of the function packet currently in process. In a multiple request packet situation, this field will be updated from first packet to second packet and so on. (See Section 4.)

3.2.2.10. End Chain Packet Address (Bytes 46-47)

This field contains the address of the last function packet to be executed. (See Section 4, Packet Advance)

3.2.2.11. Control (Byte 48)

This field contains the following control bits for the input file:

- Bit 0 (In Process) - The IP bit is set to 1 to indicate that a function is in process on the input channel. The bit is set to 1 by the issue routine and reset to 0 by the interrupt handler of the GCCR whenever a terminating interrupt (excluding the B bit interrupt) occurs.

  The GCCR normally considers a full-duplex operation if both the CBUF and TBUF keyword parameters are specified in the DTFGC macro instruction. In a half-duplex operation, this bit can be tested and output line terminal commands can be delayed until the bit is reset to zero.

- Bit 1 (End Expected) – The ND bit is set to 1 to indicate that the B bit of the buffer control word is set to 1 in anticipation of terminating the input message at the end of the current group of 64 characters. The end of any message whose length is given in the function packet is organized to terminate coincidentally with a buffer overflow.

- Bit 2 (Morphine) – The PH bit is set by the interrupt handler of the control routine to indicate that at the exit from indicator code the status in the terminating function packet was found to be '08' or '28'. When interrupt processing is resumed, the symbiont is reactivated by executing an I/O function on the communication channel.

- Bit 3 (Activity Sum) – The AS bit is set by the issue routine to indicate that the initiating of a function, other than a control function, is inhibited by finding the activity sum of the operating system too high to allow the addition of the activity value for this channel. The bit is reset in the issue routine when the activity sum is found sufficiently low to allow the execution of the function. The clock routine is employed to detect the bit and execute the issue routine.

- Bit 4 (Unused)

- Bit 5 (Timeout) – The TO bit is set by the clock routine to indicate that the entry to the interrupt routine is from the clock routine. If the bit is present, the indicator code exits to the clock routine where the bit is reset.

- Bit 6 (Unused)

- Bit 7 (Open) – The OP bit is set by the open routine if the table passes all the applied validation checks. The bit is checked in the issue routine and causes software status '12' to be set in the function packet if the bit is absent.

3.2.2.12. Receive Channel (Byte 49)

This field contains the receive channel number as it is obtained from the RCHN parameter. The number is verified against the content of the operating system unit tables during the execution of the OPEN routine.

3.2.2.13. Latest Packet Address (Bytes 50-51)

This field contains the address of the latest packet submitted to the routine. This address will be the same as that of bytes 45-46 (end chain packet) unless the packets are submitted under a multiple request. In this case, the multiple request packet will be addressed in bytes 50-51. (See Section 4.)

3.2.2.14. Indicator Code Packet Address (Bytes 52-53)

This field contains the address of the function packet in process in order to apply to the multiple request packet the address of the last packet executed under the multiple request. At the completion of the last function packet of a multiple request packet or when the chain for a multiple request is terminated for an error condition in any one of the function packets chained for execution, the address of the last function packet executed is placed into bytes 6-7 of the MRP from this field prior to the execution of indicator coding for the MRP. At the exit from indicator coding, the address of the packet being processed is restored to register 14 from this field.

3.2.2.15. Error Sense Bytes (Bytes 54-55)

This field will be set in the event of a unit check or unit exception error to the BCW status field. The execution of a "sense" XIOF will transfer the channel sense bytes to this field of the table prior to the execution of indicator coding. The sense data will only be meaningful in case of a unit check error. Tables 3-1 and 3-2 depict the first and second sense byte repertoire.

3.2.3. Output File Control Area

The output file control area is comprised of bytes 56 through 83 of the file control table and is used to control the output or transmit channel of the file. The entries in the output file control area, with the exception of the packet address and buffer address fields, parallel the entries for the input file control area. The packet address is not used in the output file control area, and the buffer address field contents are identical to the contents of the input file control area.

3.2.4. Control Area

This functional area, comprised of bytes 84 through 95, contains the dynamic fields and reference locations of the file control table. Sections 3.2.4.1 through 3.2.4.7 pertain to both input and output files.

3.2.4.1. Current Input Buffer Address (Bytes 84-85)

This field contains the address in the input buffer from which the next data transfer will take place to the input data area. The field is set initially prior to the execution of the input function and is updated after each data transfer to the address for the next transfer. The user may access this field during indicator coding to examine the data to be transferred and may modify the address in the least significant 6 bits.

3.2.4.2. Input Move Length (Byte 86)

This field contains the move length for the transfer of data from the buffer area to the input area. The field is set and changed at the same time as the preceding field (bytes 84-85) and may be modified by the use of indicator coding to alter the size of the data transfer. The value in the field is 1 less than the count of characters to be moved.

3.2.4.3. Output Move Length (Byte 87)

This field contains the move length for the transfer of data from the output area to the buffer area. The field reflects the latest move instruction until the first interrupt for buffer overflow is generated. During the interrupt processing and prior to the execution indicator coding, the field is set to the length of the move to be done. The user may alter the content of this field during indicator coding. The value in the field is one less than the count of characters to be moved.

TITLE OF SENSE BYTE	BIT CONFIGURATION P 0 1 2 3 4 5 6 7	DESCRIPTION	APPLICABLE LINE TERMINALS			
			LT-L F1003-XX	LT-M F1004-XX	LT-S F1005-XX	LT-P F1006-XX
Command Reject	P 1 0 0 0 0 0 0 0	Indicates an invalid command byte was presented to either an input or output LT.	-00, -01, -02, -03, -06,&-07	-00, -01, -02,&-03	-00, -01, -02,&-03, -04,&-05	-01
Bus Out Check	P 0 0 1 0 0 0 0 0	Indicates a parity error exists in a command byte. This error condition is detected by the LTC.	-00, -01, -02, -03, -06,&-07	-00, -01, -02,&-03	-00, -01, -02, -03, -04,&-05	-01
Data Check	P 0 0 0 0 1 0 0 0	Indicates a parity error existed on a data byte in the previous data block. This error condition is detected by the LTC and only applies to output LT's.	-00, -02, & -07	-00 & -02	-00, -02, & -04	
Overrun (Data Late)	P 0 0 0 0 0 1 0 0	Indicates data was late in being acknowledged or sent by the processor.	-01, -03, & -07	-01 & -03	-00, -01, -02, -03, -04,&-05	-01
Ring Indicator	P 0 0 0 0 0 0 1 0	Indicates that a ringing signal is being received from a remote station. A turn-on command byte must then be sent to the input LT. The DCS will answer calls automatically. This particular condition only applies to an input LT.	-03 & -07	-03	-01, -03, & -05	-01

Table 3-1. First Error Sense Byte Repertoire

TITLE OF SENSE BYTE	BIT CONFIGURATION P 0 1 2 3 4 5 6 7	DESCRIPTION	APPLICABLE LINE TERMINALS			
			LT-L F1003-XX	LT-M F1004-XX	LT-S F1005-XX	LT-P F1006-XX
Message Status (LRE) Longitudinal Redundancy Error	P 1 0 0 0 0 0 0 0	Indicates a message (or block) parity error is detected. This condition only applies to an input LT.		-03	-03 & -05	
Message Status (CPE) Character Parity Error	P 0 1 0 0 0 0 0 0	Indicates a data character (or data byte) parity error is detected in a message (or block). This condition applies to an input LT.	-03 & -07	-03	-03 & -05	
Error-Carrier OFF (AGC Lock)	P 0 0 1 0 0 0 0 0	Indicates loss of carrier when receiving a message (the INPUT DATA signal has dropped). This condition only applies to an input LT.	-03 & -07	-03	-03 & -05	
Dial No Good	P 0 0 0 1 0 0 0 0	Indicates no connection established after a dial command is sent. Only generated by an output LT connected to a Dialing Adapter (F1007-00).	-00, -02, & -06	-00 & -02	-00, -02, & -04	
Time-Out (180 ms) Break	P 0 0 0 0 0 1 0 0	Indicates the input LT has received a BREAK signal from a station on the TWX network. A BREAK signal is a spacing signal held for 180 ms duration. The BREAK signal is sent by a remote station for the purpose of stopping a transmitter. When operating in a TELEX† network, the BREAK signal indicates a disconnection.	-07			

Table 3–2. Second Error Sense Byte Repertoire

†TELEX – Trademark of Western Union Telegraph Co.

3.2.4.4. Current Output Buffer Address (Bytes 88-89)

This field contains the address in the output buffer which is the destination of the latest on imminent data transfer. The field is set at the same time as the preceding field (byte 87) and may be altered in the least significant six bits during indicator coding to change the data to be transmitted. However, aside from including an EOM character and terminating the function, there is no means to avoid transmitting 64 characters from the buffer regardless of the number of characters transferred to the buffer.

3.2.4.5. RDEVA (Byte 90)

This field is established from the value assigned to the RDEV parameter of the DTFGC macro instruction. The field is referenced in the OPEN and CLOSE coding and remains constant.

3.2.4.6. TDEVA (Byte 91)

This field is established from the value assigned to the TDEV parameter of the DTFGC macro instruction. The field is referenced in the OPEN and CLOSE coding and remains constant.

3.2.4.7. Allocation Code (Bytes 92-93)

This field contains the allocation code established by the SYMB parameter of the DTFGC macro instruction.

NOTE: Bytes 94-95 are currently unassigned but contain the revision number of the DTFGC macro instruction.

3.3. I/O FUNCTION PACKETS

I/O function packets are modules which enable the GCCR to execute input/output functions. The specific function to be performed as well as the control information required to execute that function are contained within the packet.

To initiate a function packet, the user program executes a GET or a PUT macro instruction in which the symbolic name for the function packet is specified. When executed, the GET and PUT macro instructions place a function request onto a function execution list. Once the request is made and the packet listed, control returns to the problem program while the completion of the function is pending. In order to determine the status of the function requested, the problem program checks the status bytes of the I/O function packet requested. The STATUS field of the I/O function packet is altered by the control routine as the function passes through to completion.

The format of the I/O function packet is shown in Figure 3-2.

TIME AND I/O (BYTE 0)	FUNCTION (BYTE 1)	HARDWARE STATUS (BYTE 2)	SOFTWARE STATUS (BYTE 3)
INPUT/OUTPUT AREA ADDRESS (BYTES 4 and 5)		MESSAGE LENGTH (BYTES 6 and 7)	
USER CONTROL (BYTE 8)	CONTROL (BYTE 9)	CHAIN PACKET ADDRESS (BYTES 10 and 11)	

Figure 3-2. I/O Function Packet Format

As illustrated in Figure 3-2, the I/O function packet consists of 11 bytes in which the information necessary for function execution and program control are contained. Each packet begins on a half-word boundary.

3.3.1. Time And I/O Field

The time and I/O field (byte 0) enables the problem program to specify in binary the number of seconds that the control routine waits before timing out a request for execution of a function. If this feature is not desired, the programmer must set this field to zero and must also set the T bit of the CONTROL field (byte 9) to zero.

The I/O portion of this field is determined by bit 7. This bit indicates the function to be performed with respect to the input or output channel of a device. The I/O bit is set by the GCCR as determined by the type of imperative macro instruction (GET or PUT) that requested the function packet.

3.3.2. Function Field

The function field (byte 1) contains the specific function to be performed. The contents of this field must be set by the problem program prior to the request for the packet. A repertoire of the commands presented in this field are listed in Table 3-3.

TITLE OF COMMAND	BIT CONFIGURATION P 0 1 2 3 4 5 6 7	DESCRIPTION	APPLICABLE LINE TERMINALS LT-L F1003-XX	LT-M F1004-XX	LT-S F1005-XX	LT-P F1006-XX	ACCEPTABLE MACRO INSTRUCTION
WRITE Send Data	P 0 0 0 0 0 0 0 1	A command sent during an Initial Selection Sequence which starts the Output Data Sequence and causes the LT to generate an all zeros Status Byte as a response.	-00, -02, & -06	-00 & -02	-00, -02, & -04		PUT
Dial	P 0 0 0 0 0 1 0 1	Same as Send Data. Only accepted by an Output LT equipped with a Dialing Adapter (F1007-00). Output LT does not perform the dialing function. A Turn-on Command Byte must be sent to the Input LT before sending a Dial Command Byte to the Output LT.	-00, -02, & -06	-02	-00, -02, & -04		PUT
Send Break	P 0 0 0 1 0 0 0 1	A command sent to an Output LT which causes the CI (F1002-09) to send a break (spacing) signal for a minimum of 205 ms. The command also causes the CI (F1002-10) to send a break (spacing) signal for one second. The break is normally used to stop a remote transmitter. This command causes the LT to generate an all zeros Status Byte as a response.	-06				PUT
READ Turn-On	P 0 0 0 0 0 0 1 0	A command sent during an Initial Selection Sequence which starts an Input Data Sequence and causes the LT to generate an all zeros Status Byte as a response. When data is available on the Input Data lines, it will be transmitted to the processor. This command is also used by the F1002-04, -06, or -09 CI to accept an incoming call following the receipt of the RING INDICATOR signal, or it is used to condition the CI to permit a Dial function to be performed or it is used to condition the CI to maintain the data button on the modem.	-01, -03, & -07	-01, & -03	-01, -03, & -05	-01	GET
New Sync	P 0 0 0 0 1 0 1 0	A command accepted by an Input LT which controls the CI (F1002-03 & -04 only) and enables it to quench the receive clock of a synchronous modem. This causes fast resynchronization with a newly turned-on remote transmitter and causes the Input LT to generate an all zeros Status Byte as a response. This command is normally used for multiple-party connections only.			-01, -03, & -05		GET

Table 3–3. Command Byte Repertoire (Part 1 of 3)

TITLE OF COMMAND	BIT CONFIGURATION P 0 1 2 3 4 5 6 7	DESCRIPTION	APPLICABLE LINE TERMINALS				ACCEPTABLE MACRO INSTRUCTION
			LT-L F1003-XX	LT-M F1004-XX	LT-S F1005-XX	LT-P F1006-XX	
Look-For-Sync (LFS) or Parallel Test	P 0 0 0 0 0 1 1 0	A command which nullifies character synchronization in an Input LT-S and then causes the LT-S to identify two contiguous unique Sync characters and a nonsync character (SOM character is optional out of a serial data stream. New character synchronization is then established. This command also causes the Input LT-S to stop sending data. The LT-S begins sending data only when a non-sync character (SOM or Data character) is received following re-synchronization. This command causes the input LT-S to generate an all zeros Status Byte as a response. Parallel Test provides the LT-P with back-to-back testing capability (used only during the DCS Test mode). It is the final command transmitted by the processor and immediately follows a DCS Test and a Turn-On command.			-01, -03, & -05	-01 (Maintenance Only)	GET
Answer Back A Answer Back B Answer Back AB	P 0 0 0 1 0 0 1 0 P 0 0 0 1 0 1 1 0 P 0 0 0 1 1 0 1 0	A command sent to the Parallel LT which allows the CI (F1002-07) to send three different tones on the ANSWER-BACK channel of the 403D5 or 403D6 Data Set. The processor can select and control these tones by means of the coding within the command. The select-tone is sent for three to five seconds. This command causes the Parallel LT to generate an all zeros Status Byte as a response.				-01	PUT
CONTROL Turn-off	P 0 0 0 0 0 0 1 1	A command which causes the immediate termination of an Input or Output Data Sequence and causes the LT to generate a Channel End - Device End Status Byte as a response.	-00, -01, -02, -03, -06,&-07	-00, -01, -02&-03	-00, -01, -02, -03, -04,&-05	-01	PUT and GET
DCS Test	P 0 0 0 0 1 0 1 1	A command to the Input LT which terminates the connections between the CI and the communication lines or modem and causes the Input LT to generate a Channel End - Device End Status Byte as a response. This command connects the output signal to the input signals for loop-back testing of the DCS by the processor.	-01, -03, & -07	-01 & -03	-01, -03, & -05	-01	GET

Table 3–3. Command Byte Repertoire (Part 2 of 3)

TITLE OF COMMAND	BIT CONFIGURATION P 0 1 2 3 4 5 6 7	DESCRIPTION	APPLICABLE LINE TERMINALS LT-L F1003-XX	LT-M F1004-XX	LT-S F1005-XX	LT-P F1006-XX	ACCEPTABLE MACRO INSTRUCTION
End Test	P 0 0 0 0 1 1 1 1	A command to the Input LT which switches the CI back to normal operation following a DCS or Local Test operation and causes the Input LT to generate a Channel End - Device End Status Byte as a response.	-01, -03, & -07	-01 & -03	-01, -03, & -05	-01	GET
Disconnect	P 0 0 0 1 0 0 1 1	A command to the CI (F1002-04,-06, or 09) by means of the Input LT for the purpose of terminating a call. Upon receipt of the signal, the DATA TERMINAL READY signal is dropped to the modem, indicating a "hang-up". This command causes the input LT to generate a Channel End-Device End Status Byte as a response.	-01, -03, & -07	-01 & -03	-01, -03, & -05	-01	GET
Local Test	P 0 0 0 0 0 1 1 1	A command to the Input LT which causes the CI (F1002-05) to exercise the Local Test Control on the associated modem. This command places the modem in the loop-back test mode for testing all local (on-site) hardware including the modem. (Applies only to modems which have turn-around capability.) This command causes the Input LT to generate a Channel End - Device End Status Byte as a response.	-01, -03, & -07	-01, & -03	-01, -03, & -05		GET
SENSE	P 0 0 0 0 0 1 0 0	When an error condition exists in the DCS, the appropriate line terminal sends (by way of the LTC) a Unit Check Status Byte to the processor. The processor then replies with a Sense command. The Sense command tells the line terminal to return an all zeros Status Byte followed by two Sense Bytes.	-00, -01, -02, -03, -06,&-07	-00, -01, -02,&-03	-00, -01, -02, -03, -04,&-05	-01	PUT and GET
TEST I/O	P 0 0 0 0 0 0 0 0	A command which is used to obtain the present status of a line terminal. This command does not cause the generation of new status but does cause the LT to generate an all zeros Status Byte as a response.	-00, -01, -02, -03, -06,&-07	-00, -01, -02,&-03	-00, -01, -02, -03, -04,&-05	-01	PUT and GET

Table 3-3. Command Byte Repertoire (Part 3 of 3)

3.3.3. Status Field

The status field (bytes 2 and 3) contains a hardware status field (byte 2) and a software status field (byte 3). The contents of these two fields express the status of function controlled by the packet as presented by the hardware and software required to perform that function. The GCCR provides this information to inform the problem program of the status of the function packet. The manner in which contents of this field is determined is illustrated in Figure 3-3 and described in the paragraphs that follow.

When a request by means of a GET or PUT macro instruction is accepted by the GCCR, byte 3 is set to 'FD' to indicate that the command has been placed on a queuing list and will be executed as soon as facilities are available. The coding of byte 2 is unaffected at this time and remains at '00'. Prior to issuing a command to a line terminal, a test is made to check the hardware status of the subchannel assigned to the line terminal. The purpose of the test is to determine whether the command can be issued or if it must be delayed. If a delay is required, byte 2 is set to '10' (busy condition) and the GCCR will continually attempt to reissue the command. Control, however, will not be given back to the user until the busy line terminal is ready to accept the command or a command rejection occurs. If the command is rejected, the status byte is set to reflect the condition which caused the rejection. For example, status byte coding of '21' shows that rejection was due to a data or parity error condition while a status byte coding of '22' indicates that a nonoperational or off-line device was addressed.

When a command passes the initial selection sequence (accepted by the channel), a status code of 'FE' is marked in byte 3. The status coding of byte 2 remains at '00'. Whenever a command fails the initial selection sequence (command rejected), byte 2 is set to '80' and a terminating interrupt occurs or a clock times out. The setting of byte 2 to an '80' code allows the user to determine when his function is completed. To determine the true ending condition of the command, byte 3 must also be examined when byte 2 is set to '80'.

If the problem program employs indicator coding, then the indicator code uses the contents of the software status byte to direct the activity of the GCCR in the handling of specific status conditions. These conditions are presented along with the conditions for other software status codes in Table 3-4. The codes marked with an 'I' in the restriction column of this table are set by user indicator code as directives to the GCCR. The codes marked with an "S" in the restrictive column are symbiont control directives. All other codes are set by the GCCR.

3.3.4. Address Field

The address field (bytes 4 and 5) contains the address of the area for the data of the I/O function. The address is provided to the GCCR when a data transfer is a requirement of the function requested. If no data transfer takes place, the programmer must set the contents of the field to zero.

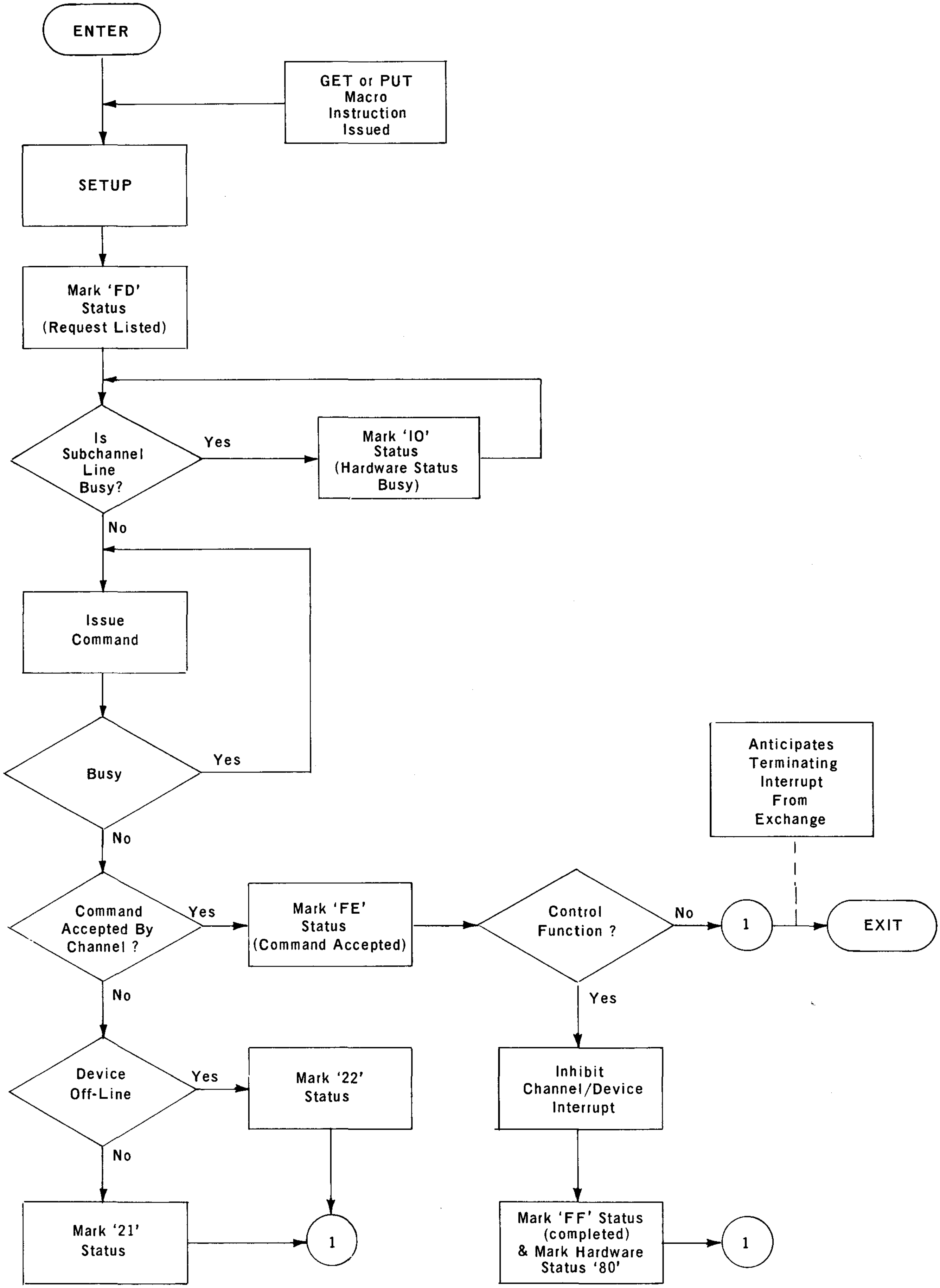

Figure 3-3. Initial Command Selection Sequence

STATUS CODE	RESTRICTION	MEANING
		TERMINATING INTERRUPTS (LT EFFECTIVELY TURNED OFF)
01		Unit exception has occurred (unit offline).
02		Unit check has occurred (sense data available in file control table).
03		Memory protection termination – data transfer was truncated to prevent main storage distortions. If data accesses byte 64 or 128 of the modulo 128 dynamic buffer (CBUF/TBUF) and a buffer segment interrupt (status code 11) was already present, the hardware automatically truncates the data transfer and the line terminal is turned off.
FF		Command successfully completed.
		INITIAL SELECTION SEQUENCES (COMMAND ISSUANCE)
10		Device requested is busy – this status is temporary and the user will never get control when this status is set. It does not imply to reissue the request. Prior to issuing a command, a test is made to see if the device is busy. If busy, the status '10' is marked. However, the dispatcher attempts to reissue the command until an acceptance or rejection of the function is set at the initial selection sequence.
21		Command rejected because of a data or parity check – this condition is normally set by a parity error in a device address or a parity error in a command function. The user should design his program to set limits on the successive number of times that a command should be reissued before aborting. A display should be incorporated with the abort of the program.
22		Command rejected because of addressing a nonoperational device – this condition can occur if the off-line switches on a DCS-1 or DCS-4 are set. It can also be indicative of addressing a nonexistant device. A display to indicate that the unit is offline should be included in the user's problem program.
FD		Command listed – the dispatcher has placed the command on a queueing list and will execute the command as soon as facilities are available.
FE		Command successfully initiated – the command has passed the initial selection sequence and has been accepted by the channel. Terminating interrupts can be expected if this is not a control command. Control commands completed are marked successfully completed at initial selection sequence.
		CLOCKING INTERRUPTS
06		Function has timed out. If no indicator coding is associated with this command, the dispatcher will issue a TURN OFF to the line terminal. If indicator coding is used but does not manipulate the file control table or function packet, the dispatcher will issue a TURN OFF to the line terminal. If indicator coding only changes the status to '04', this function will be given no time limits and will never time out or be turned off. If indicator coding changes the status to '04' and manipulates the input or output time countdown field of the file control table, the line terminal will not be turned off until the user allows the time to be counted down and does not reissue as illustrated. If indicator coding does not change the '06' status but changes either the input or output count down field to a nonzero positive value, the timing of the command will continue without a turn off issued to the appropriate line terminal and without reissuing the command. It actually disregards the initial timeout and continues counting down.

Table 3–4. Status Byte Coding for I/O Function Packet (Part 1 of 2)

STATUS CODE	RESTRICTION	MEANING
		MISCELLANEOUS CONDITIONS
04	I	Permits the same function to continue (see '06' status code).
05	I	Terminates current function and does not initiate new function. Turns off the appropriate line terminal and resets all packet address fields in the file control table to zero.
08	I S	Symbiont release – reissue same function when routine becomes active again.
09	I	Do not advance list to next function at this time but reissue the current packet.
11		Buffer segment interrupt – the dynamic buffer is divided into two half buffers for hardware addressing. Whenever the limits of the upper half buffer (byte 64) are accessed, an interrupt occurs and data is allowed to move into or out of lower half buffer. If an output function is in force, the interrupt handler will unload the upper half buffer to a user specified area while the data is inputting to the lower half buffer. If an output function is in force, the interrupt handler will load the upper half buffer with a consecutive data stream while the lower half buffer is outputting. The buffer segment interrupt will also occur when the limits of the lower half buffer (byte 128) is accessed. The next location allowed to be accessed is byte 1 of the upper half buffer. The procedures are the same as for upper half buffer interrupts. (See 03 status code.) The handling of this condition is automatic.
12		File is not opened and function will not be executed.
13	S	Function request is dormant awaiting reactivation I/O activity by operating system.
28	I S	Defer function execution until I/O activity reactivated by operating system.

Table 3–4. Status Byte Coding for I/O Function Packet (Part 2 of 2)

3.3.5. Message Length Field

Bytes 6 and 7 comprise the message length field. If a data transfer is a requirement of the function request, then the user program enters into this field the binary representation of the message length. For messages containing an end-of-message character (EOM), the value specified includes the EOM character. If the user wants to determine the length of the message by an EOM character, then the contents of the field must be set to zero.

3.3.6. User Control Field

The user control field (byte 8) is reserved to allow the problem program to communicate with its indicator coding. The contents of this field are not altered by the GCCR.

3.3.7. Control Field

The control field (byte 9) is comprised of eight control bits providing coordination between the problem program and the GCCR. The problem program must set these bits to a binary 1 condition to indicate the condition and the action to be taken by the GCCR.

Table 3-5 describes the significance of each bit. Examples of typical usages are provided in Section 5.

BIT NUMBER	MEANING
0	When set to 1, indicates function packet is a multiple request.
1	When set to 1, indicates that indicator code is to be executed for error conditions which arise from execution of I/O functions.
2	When set to 1, indicates that indicator code is to be executed for the successful completion of the I/O function.
3	When set to 1, indicates that the function is a control function rather than a data transfer function.
4	When set to 1, indicates that time field (byte 0 of packet) is to be used to limit time allotment for function execution.
5	When set to 1, indicates indicator code is to be executed when time limit specified by byte 0 of packet has been exceeded.
6	When set to 1, indicates that indicator code is to be executed each time the buffer control word interrupt occurs during the transfer of data to or from a communications line.
7	When set to 1, indicates that the function is to be executed on the transmit channel. When zero, indicates that function is to be executed on the receive channel. This bit must be set by the problem program for those function packets submitted under a multiple request.

Table 3–5. Control Byte Coding for I/O Function Packet

3.3.8. Chain Field

The chain field (bytes 10 and 11) chains or links function packets together when more than one function packet is required to accomplish a function. The GCCR uses the field to chain packets when more than one packet is requested or if a new packet is requested, before the previously requested packet has been completed. If the user program chains a sequence of packets, then the chain field of each packet, with the exception of the last packet, must specify the address of the next packet in the sequence to be executed.

3.3.9. Typical Function Packet Coding

The coding for typical output and input function packets are shown in Figures 3-4 and 3-5, respectively.

```
LABEL    OPERATION  OPERAND                                COMMENTS
         DS         0H
SEND     DC         YL1(0)         DO NOT TIME OUT
         DC         X'01'          FUNCTION IS SEND DATA
         DC         Y(0)           INITIAL HARDWARE SOFTWARE STATUS
         DC         Y(OUTB)        OUTPUT WORK AREA
         DC         Y(0)           NO LENGTH SIGNIFIES EOM TERMINATION
         DC         X'0'           USER CONTROL
         DC         X'61'          THIS PACKET IS NOT A MULTIPLE REQUEST
*                                  ANY TERMINATING ERROR GIVES CONTROL TO
*                                  OUTPUT INDICATOR CODE
*                                  ANY SUCCESSFUL COMPLETION GIVES CONTROL
*                                  TO OUTPUT INDICATOR CODE
*                                  THIS IS A DATA EXCHANGE AS OPPOSED TO A
*                                  CONTROL FUNCTION
*                                  THERE IS NO TIMING OF THIS FUNCTION
*                                  THIS IS AN OUTPUT CHANNEL FUNCTION
```

Figure 3-4. Typical Output Function Packet Coding

```
LABEL    OPERATION  OPERAND                                COMMENTS
         DS         0H
TURNON   DC         YL1(1)         TIME OUT IN 1 SECOND
         DC         X'02'          FUNCTION IS TURN ON INPUT LINE TERMINAL
         DC         Y(0)           INITIAL HARDWARE/SOFTWARE STATUS ZERO
         DC         Y(INWK)        INPUT WORK AREA
         DC         Y(128)         MESSAGE LENGTH PLUG VALUE OTHER THAN 0
         DC         X'0'           USER CONTROL
         DC         X'80'          BIT 4 SET TO 1 TO INDICATE TURN OFF INPUT
*                                  WHEN TIMED OUT BIT 7 OFF INDICATES INPUT
*                                  FUNCTION

*                   THIS PACKET IS NORMALLY THE INITIAL PACKET ISSUED UNDER
*                   A GET. THE TIMEOUT OF 1 SECOND IS DELIBERATE TO MAKE SURE
*                   THAT THE INPUT LINE TERMINAL IS OFF BEFORE ISSUING A
*                   PUT COMM. THIS PACKET DOES NOT EXPECT INPUT DATA AND IS
*                   EXPECTING TO ISSUE A SEND DATA UPON TERMINATION OF THIS
*                   PACKET.

INWK     DS         CL128
```

Figure 3-5. Typical Input Function Packet Coding

3.4. MULTIPLE REQUEST PACKET

Multiple request packets are a special form of I/O function packets which allow the problem program to submit a request for a sequence of I/O functions through the execution of a single GET or PUT macro instruction. The GCCR recognizes a multiple request packet by a bit present in the most significant bit position in the control field (byte 9) of the packet. When set, this bit instructs the GCCR to set a sequence for the I/O function packets to be executed. The sequence is established by storing into the last halfword of each function packet the address of the next function packet in the sequence. The sequence is introduced to the GCCR by specifying the address of the first and last I/O function packets of the sequence within the multiple request packet.

The format of the multiple request packet is illustrated in Figure 3-6. The format of this packet is similar to the format of the I/O function packet. The time field, function field, and message length field however do not apply for the multiple request packet, while the status fields and user control field remain the same for both types of packets.

<table>
<tr><td colspan="2">NOT REQUIRED
(BYTES 0 and 1)</td><td>HARDWARE STATUS
(BYTE 2)</td><td>SOFTWARE STATUS
(BYTE 3)</td></tr>
<tr><td colspan="2">FIRST FUNCTION PACKET ADDRESS
(BYTES 4 and 5)</td><td colspan="2">NOT REQUIRED
(BYTES 6 and 7)</td></tr>
<tr><td>USER CONTROL
(BYTE 8)</td><td>CONTROL
(BYTE 9)</td><td colspan="2">LAST FUNCTION PACKET ADDRESS
(BYTES 10 and 11)</td></tr>
</table>

Figure 3-6. Multiple Request Packet Format

Two fields differing from the I/O function packets are those specifying the address of the first function packet (bytes 4 and 5) and the last function packet (bytes 10 and 11) to be executed in the sequence. Note that only the first three bits of the control field (byte 9) are used by the multiple request packet. The use of these three bits is the same as that specified for the first three bits of the control field for I/O function packet.

3.5. INPUT AND OUTPUT DATA TRANSFERS

Separate explanations for input and output data transfers are provided.

3.5.1. Input Data Transfer

The function packet contains an input area address which is used as a destination address of data transfers originating from the CBUF. The transfer of data from the CBUF to a user specified work area takes place whenever a terminating interrupt (successful or error) or a buffer segment interrupt occurs. Time out conditions will not deliver data from CBUF to work areas. Data may or may not have entered CBUF if an asychronous terminal was used. Indicator coding could be used to check this condition. If a message of 80 characters was successfully received and a length of zero was specified, two transfers to the user area would take place. The first 64 bytes would be transferred when a buffer segment interrupt caused by accessing the limits of the upper half buffer occurs. The second transfer would be a 16-byte transfer from the lower half buffer when the successful completion interrupt occurs. The length specification of zero is indicative of line terminal capability of termination based on EOM recognition. The GCCR conditions itself to receive the first byte of data left-justified in the upper half buffer when a length of zero is specified. The GCCR conditions itself to receive the last byte of a message right-justified on either the 64^{th} byte or 128^{th} byte of CBUF whenever a value other than zero is specified as a message length. This technique is used to force a memory protection termination interrupt for line terminals or specific messages that can not end with EOM recognition. A line terminal capable of ending with EOM recognition can have a true or maximum message length specified; it is not mandatory to specify zero. The EOM character recognition would be coincidental with the memory protection interrupt if a true message size was specified.

Specifying a zero length message for EOM recognition purposes is not recommended. A typical dangerous situation can exist if the remote terminal was transmitting from a paper tape reader and EOM was not present in the tape. The GCCR has no maximum message size to monitor and to use to turn off the input line terminal.

If the problem program has an entry to indicator coding based on a B bit setting in the control field of the function packet, control will be given to indicator coding whenever a buffer segment interrupt occurs and prior to the transfer from CBUF to the user work area. Counting the number of B bit interrupts in indicator coding and marking the software status byte to a ‘05’ (terminate current function and do not initiate new function) can prevent memory runaway when zero is used as message size. The user can then mark the User Control byte of the function packet to control his recovery procedures and to notify the remote terminal operator.

3.5.2. Output Data Transfer

The address field of a function packet is used as an origin address in output data transfers to a remote terminal. The CBUF/TBUF receives the data from the work area specified in the function packet and transmits the data. If the message length is zeros, the GCCR moves the data from the user defined work area to the upper half of the dynamic buffer left-justified. The specification of zero message length implies that an EOM character is the last byte of the message and that the output line terminal is capable of termination based on recognition of this character.

Where EOM recognition capability does not exist in an output line terminal or where specific messages are not framed by an EOM character, the user must specify the exact message length excluding hardware generated control characters, such as SYN, CRC, LRC. The GCCR will right-justify the last byte of the message on either the 64th or 128th byte of the dynamic buffer to force a memory protection termination where an explicit message length is specified. To terminate where EOM recognition is not in force, it is imperative that the true message length be specified. If a true message length is specified and EOM termination exists, the message will terminate coincidently based on both EOM detection and memory protection interrupt.

Each time a buffer segment interrupt occurs, the GCCR will load the interrupting half buffer with 64 bytes from the data stream specified in the user work area and will erase the B bit in the BCW. If GCCR is anticipating an EOM character in the message and this character is not embedded in the data, all characters in memory will be transmitted until an addressing error occurs. This condition will be unrecoverable if the problem program contains indicator coding and the B bit is set in the control field of the function packet indicator coding will be executed at each B bit interrupt prior to the data transfer required for that interrupt.

4. PACKET ADVANCE

4.1. GENERAL

Packet advance is the process of updating the file control table of the GCCR to include the information necessary for the execution of an I/O function packet. Each time the execution of an I/O function packet has been completed and a new packet is scheduled for execution, the GCCR updates the file control table with information pertinent to the new packet. The two areas of the file control table updated are the input file control area and the output file control area. (Refer to Figure 3-1 for an illustration of the file control table format.) The specific area updated is determined by the type of function to be performed (input or output) and the systems communication mode (half-duplex or full-duplex). The correlation between these two factors and the area of the file control table affected is illustrated in the following matrix. This matrix will depict the area chosen to list First Packet Address, End Chain Address, and Latest Packet Address fields.

MODE OF OPERATION	FUNCTION TYPE	REQUESTING MACRO	FILE CONTROL TABLE AREA AFFECTED
FULL DUPLEX	INPUT	GET	INPUT
	OUTPUT	PUT	OUTPUT
HALF DUPLEX	INPUT	GET	INPUT
	OUTPUT	PUT	INPUT

The important fact to remember is that the input area of the file control table is utilized for both input and output operations in half-duplex mode. (Half duplex is denoted by specifying only the CBUF parameter (TBUF omitted) in the DTFGC declarative macro instruction; refer to 2.7.)

Since the fields of the file control table are duplicated for both input area and output area, the discussions and examples used in this section will refer to the general name of the field followed by the byte numbers of each area. For example, Time Countdown field (bytes 32 or 60); the appropriate byte is determined by the matrix previously discussed.

4.2. PACKET ADVANCE (EXCLUDING PACKET CHAINING)

When chaining is not a factor in packet advance, the GCCR is primarily concerned with moving only three fields of the I/O function packet to the appropriate area (input or output) of the file control table. The time and I/O field (byte 0) is moved to the time countdown field (bytes 32 or 60) of the control table for time countdowns. The packet message length (bytes 6-7) is moved to the Message Length Countdown field (bytes 34 or 62) of the control table for message length countdowns. The input/output area address is moved from bytes 4-5 of the packet to the Current Area Address field (bytes 30 or 58) of the control table to mark the user work areas.

Each time a new I/O function packet is scheduled for execution, the process is repeated. This process applies to all I/O function packets whether they are initiated from within the user program by a macro call or advanced as a result of packet chaining. The latter, however, requires additional information to be provided to the file control table and is therefore discussed separately in this section.

4.3. PACKET ADVANCE (INVOLVING PACKET CHAINING)

Packet advance can be initiated by a method referred to as chaining. Packet chaining is employed when more than one I/O function packet must be executed to perform a specific function or if the user desires to sequence a group of function packets for execution. Chaining can be accomplished by two methods: by utilizing the Chain Packet Address field of the I/O function packets or by requesting the execution of a multiple request packet. Although both methods accomplish basically the same thing, the manner in which the GCCR handles packet advance and the information which must be supplied to the GCCR differs.

4.3.1. Packet Chaining In I/O Function Packets

Packet advance occurs the same as described in 4.2 for each I/O function packet executed by the GCCR. However, the sequencing of packets to be advanced is accomplished through the use of the Chain Packet Address field (bytes 10 and 11) of an I/O function packet. Figure 3-2 illustrates the format of an I/O function packet. The chaining field is used by both the GCCR and the user program. The GCCR uses the field to chain packets when more than one packet is required to accomplish a function or when a new packet is presented before the current packet is completed. The user program uses the chaining field for sequencing the execution of a group of packets.

When chaining is specified, the contents of the chaining field (address of next packet to be executed) is moved from the I/O function packet into the End Chain Packet Address of the file control table. Upon completion of the packet currently being executed, the packet address contained in the Chain Packet Address field is moved to the First Pack Address field in the file control table. The GCCR has now advanced to the next packet in the sequence to be executed. If an additional packet is to be executed in this sequence,then the chaining field of the newly scheduled packet must specify the address of the next packet to be executed. The process repeats itself until the last function packet of the sequence is reached. The last packet must not have an entry in the chaining field. When the GCCR completes the execution of the last function packet, control is returned to the user program.

4.3.2. Packet Chaining In Multiple Request Packets

The multiple request packet is a special form of I/O function packet which permits the user to submit a request for the execution of a sequence of I/O function packets through the execution of a single GET or PUT macro instruction. The information contained in the multiple request packet provides the GCCR file control table with the address of the first function packet to be executed, the control information necessary to direct the actions of the GCCR, and the address of the last function packet to be executed. The function packets existing between the first and last packets specified in the multiple request packet are linked through the use of the chain packet address in each of the I/O function packets. (See 4.3 for information concerning the use of the Chain Packet Address field in I/O function packets.)

When the request for the multiple request packet is executed, the address of the packet is loaded into the Latest Packet Address of the file control table. The first function packet address and the last function packet address specified in the multiple request packet are loaded into the First Packet Address and the End Packet Address fields of the file control table, respectively. The addresses in these fields are changed (updated) upon completion of each packet so that the next packet for execution is presented to the GCCR and the next packet chained to it is listed. The GCCR determines the completion of multiples requests by comparing the address contained in the First Packet Address field of the file control table to the address in the End Packet Address field of the table. When these two addresses are equal and the function packet completed, the GCCR considers the multiple request completed. Control is then returned to the user program. It is important not to repeat packet addresses within the chaining sequence; otherwise packets may be aborted.

Figure 4-1 illustrates how the function packet addresses (A, B, C, and D) are advanced within the fields of the file control table as each of the sequentially chained function packets are completed.

EVENT	GCCR FILE CONTROL TABLE FIELDS		
	FIRST PACKET ADDRESS	END PACKET ADDRESS	LATEST PACKET ADDRESS
Execution of Multiple Request Packet (Initial Setting)	PACKET A	PACKET D	PACKET M (Address of Multiple Request PK)
Execution of PACKET A completed	PACKET B	PACKET D	PACKET M
Execution of PACKET B Completed	PACKET C	PACKET D	PACKET M
Execution of PACKET C Completed	PACKET D	PACKET D	PACKET M
Execution of PACKET D Completed	ZERO	ZERO	

Figure 4-1. Packet Advance Sequence for Multiple Requests

4.3.2.1. Half-Duplex Mode Multiple Request Considerations

Function packets for both input and output functions may be chained and submitted for execution through a multiple request packet. Prior to chaining of the function packets, the user must perform the following:

(a) The I/O bit (bit 7) of the control byte (byte 9) must be set to indicate whether the function is to be executed on the input channel or the output channel. If the function is to be executed on the input channel, the I/O bit must be set to 0. The I/O bit must be set to '1' if the function is to be executed on the output channel.

(b) The Chain Packet Address field (bytes 10 and 11) must specify the address of the function packet to be executed upon the successful completion of this packet. The chaining of the sequence of function packets is initiated by executing either a GET or PUT macro instruction in which the address of the multiple request packet is specified as the second parameter:

```
   GET  filename, mrpaddress
or PUT  filename, mrpaddress
```

4.3.2.2. Full Duplex Mode Multiple Request Considerations

In full-duplex mode, the sequence of function packets may be chained and submitted for execution in the same manner as described in 4.3.2.1. However, the setting of the I/O bit in the control field of each function packet depends upon the particular macro instruction through which the function is submitted. The I/O bit must be set as described in 4.3.2.1 for functions submitted through a PUT macro instruction. Functions submitted through a GET macro instruction *must not* contain an entry in this bit position.

4.4. THE EFFECT OF CONTROL FUNCTIONS ON PACKET ADVANCE

Packet advance for a group of chained packets can be terminated or unaffected by the execution of a control function. The action taken is determined by the method in which the control function is submitted for execution. Control functions are contained in I/O function packets and are distinguishable from data transfer functions by presence of the control bit (bit 3) of the control byte (byte 9) in the packet. These functions may be submitted for execution through GET or PUT macro instructions or as part of a group of chained packets through a multiple request packet. The method selected, however, will have certain effects upon other function packets also listed for execution of the selected file. For example, a control function submitted for execution through a GET or PUT macro instruction to preempt a chain of function packets for a given file is executed immediately by the GCCR. The execution of the control function will terminate the list of function packets awaiting execution for that file, thus prohibiting packet advance and execution. If, however, the control function was submitted as one of a group of function packets through a multiple request packet, it would be executed in its proper sequence and would have no effect upon the function packets preceding it or upon the advancement and execution of the function packets following it.

4.5. THE EFFECT OF INDICATOR CODINGS ON PACKET ADVANCE

Through the use of indicator coding, a user may initiate directives to the I/O dispatcher and control routine which directly affect packet advance. The three directives involved are the '09', '05', and '04' status codes listed in Table 3-4. By marking the software status field of the function packet in force with the appropriate status, the user can direct packet advance if the normal processing of the function is interrupted. For example, if the software status field of the function packet in force is marked with an '09' status and an interrupt occurs, the GCCR is effectively directed to reissue the GET/PUT macro instruction for that function while in the I/O mode. An '05' status marked in the software status field enables the user to cancel a function in process and any function packets associated with this packet through chaining. To override a time out condition and the subsequent timing of a line terminal, the user need only specify an '04' status in the software status field of the in force function packet. This status code will allow the current function to continue although it has exceeded the specified time out entry.

5. ERROR CONDITIONS AND RECOVERY

5.1. GENERAL

Certain error conditions require operator action at the central processor. When these conditions exist, a display results at the control console of the processor. Some conditions causing the error display stops are recoverable; others are not recoverable. In cases where indicator coding is specified, the GCCR is directed as to the course of action to be taken. This action may or may not require operator intervention. The aspects of error conditions and error recovery are discussed in this section.

5.2. ERROR RECOVERY INVOLVING OPERATOR INTERVENTION

In general, the operator interface will be the responsibility of the problem program. The control routine will make available to the problem program the status of the functions executed, and the problem program will interpret the status and determine its own course of action which may or may not include operator intervention.

The OPEN macro instruction executes coding which sets and verifies the allocation of devices assigned to the routine. The operator will be notified of the error conditions by means of display stops. Those displays which are recoverable allow the operator to keyin a response which forces the control routine to accept the conditions as they are found and to continue to process as though the error had not been encountered. The displays which are not recoverable provide no alternatives for the operator. Any response to those displays will result in the same action as a negative response to the marked displays. That action is to inhibit setting the file OPEN. The control routine will proceed through the remainder of the validity checking of the OPEN coding and make additional displays as appropriate but will exit without setting the OPEN indicator bit for the file in error.

When a function packet is presented to the control routine for the unopened file, the STATUS byte of that packet will be set to notify the user routine that the error condition exists. The user routine must then avoid using the unopened file until the required correction is made.

Table 5-1 and 5-2 list recoverable and nonrecoverable display stops.

HEXADECIMAL DISPLAY	CAUSE
75xx	The allocation code given the GCCR through the SYMB parameter of the DTFGC macro instruction does not agree with the allocation code listed in the PU table entry for the file being opened.
7Axx	The device requested is not operable.

NOTE: xx represents the input or output channel number.

Table 5–1. Recoverable Display Stops

HEXADECIMAL DISPLAY	CAUSE
76xx	Device specified by DTFGC is not a communications device.
7Cxx	The communications buffer specified by DTFGC macro instruction does not lie on a 128-byte boundary (modulo 128).
78xx	LUPU table entry disagrees with channel specified in DTFGC macro instruction.

NOTE: xx represents the input or output channel number.

Table 5–2. Nonrecoverable Display Stops

5.3. USE OF INDICATOR CODING FOR ERROR RECOVERY

The purpose of indicator coding is to alert the problem program of changes in the I/O function status due to error conditions and to direct the GCCR of the error recovery to be initiated. When indicator coding is specified, the occurrence of an error condition executes the indicator code as a subroutine of the interrupt handling function of the GCCR. The execution of the indicator code is performed in the I/O mode of operation. The use of indicator is accomplished under the following restrictions:

- The user returns control to the routine in a minimal amount of time. The coding records the error condition for which the indicator coding is being executed. The processing of the information recorded is deferred until the problem program is returned to the processor mode.
- The problem program must not execute any imperative macro instructions for this routine or any other routine within the indicator coding.
- The indicator coding is not to use I/O register 8. Registers 11 through 15, however, can be used to interface the GCCR with the indicator coding. The contents of registers 11 through 15 are preserved by the indicator coding.

5.3.1. Use Of I/O Registers

The contents of the I/O registers, at the time that indicator coding is executed, contain parameters which are used in relating the initiating of the indicator coding and the information necessary to permit the indicator code to exit. The contents of each I/O register are as follows:

- I/O register 11

  This register contains the address of the first character of the function packet responsible for the interrupt. The status field of that packet provides the reason for the entry into the indicator coding.

- I/O register 12

  This register contains the address of the first character of the input or output section of the file control table for the specific file whose interrupt is being processed.

- I/O register 13

  This register contains the address of the first character of buffer control word which controls the communication device for the file.

- I/O register 14

  This register contains the return address to which the indicator exits.

- I/O register 15

  This register contains the address of the first character of the packet control section of the file control table.

5.3.2. Options Available To The Problem Program

The problem program has the option to direct the action of the GCCR from the conditions it detects in the indicator code. There are three options:

- Continue to process with no further corrective action;
- Determine to clean up in preparation to cancel the problem program;
- Manipulate the listed function packets to alter the further processing by the GCCR.

The options offered will be put into effect by accessing the file control table and the function packets, and by communicating the option elected to the GCCR at the exit from the indicator code.

5.3.3. Entry And Exit Of Indicator Coding

Since the problem program indicator coding is treated as a subroutine, it is entered by the execution of a branch and link instruction. The format of the BAL instructions is:

```
BAL 14,0(,14)
```

where 14 is the I/O register which contains the address to which the indicator coding exits upon completion.

When completed, the indicator coding executes the following instruction in order to exit back to the GCCR:

```
BC 15,0(,14)
```

5.3.4. GCCR Action Resulting From Indicator Code Status

Once the indicator code exits to the GCCR, the status byte of the indicator coding is examined so that the action to be taken by the GCCR can be determined. The status conditions and the GCCR actions taken are as follows:

- '04' Status *Maintain Function*

  If at the exit from indicator coding the status byte is found to be set to '04', the control routine will avoid those activities which relate to the termination of a function. These activities include:

  - The activity sum will be left unchanged indicating that data transfer from or to the remote device may be anticipated.
  - The list of function packets is not advanced; the current function packet remains current.
  - The issue routine is not executed.

- '05' Status *Cancel*

  If at the exit from indicator coding the status is found to be set to '05', the control routine will terminate the current function and will execute a "turnoff" function to the communication channel involved.

- '06' Status *Timeout*

  If at the exit from indicator coding the status is found to be set to '06', the control routine will determine its action from the content of the "time countdown" field of the input/output section of the file control table, the address of which is specified by register 12. If the time countdown remains zero, the action of the control routine will be the same as that for '05' status. If the content is a positive binary number, the action will be that for '04' status.

- '08' Status *Symbiont Release – Inhibit List Advance*

  If at the exit from indicator coding the status byte is found to be set to '08', the control routine will retain the current function packet and suspend the execution of the function as described for '28' status.

- '09' Status *Inhibit List Advance*

  If at the exit from indicator coding the status is found to be set to '09', the control routine will terminate and reissue the function specified in the current function packet. Actions taken include:

  - The activity sum will be reduced.
  - The list of function packets is not advanced.
  - The issue routine is executed.

■ '28' Status *Symbiont Release*

Status '28' is intended to allow a symbiont to order the control routine to suspend activity on a communications device while the operating system performs a function which requires that all I/O activity be suspended. The symbiont indicator coding satisfies this requirement by chaining a function packet containing a '28' status to the current function packet. At exit from the indicator coding, the new current function packet is determined and examined for the '28' status. If the '28' status is detected, the control routine will suspend the execution of the issue routine for that function until the operating system indicates the reactivation of I/O activity by reentering the interrupt handling routine. The suspended function is then executed and the I/O cycle is resumed.

6. PROGRAMMING CONSIDERATION

6.1. GENERAL

This section shows the standard procedures required for terminals that communicate with a UNIVAC 9200/9300 central processor site. When the primary direction of message traffic is from the central processor to the remote terminal, the mode of operation is an output or message transmission mode and is referred to as a transmit program. When the direction of message traffic is primarily from a remote terminal to the central processor, the mode of operation is an input or message receive mode and is referred to as a receive program.

6.2. INITIALIZING A MANUAL DIAL RECEIVE PROGRAM

After all required peripheral devices and the communications devices have been opened, the first command issued to the DCS-1/DCS-4 from the central processor is a GET macro instruction with a TURN-ON function (02_{16}) . This command conditions the modem to prevent the DATA pushbutton indicator on the hand-phone from turning off once it has been pressed. If only one remote terminal is used, a halt display can be issued immediately by the user program after the GET instruction and TURN-ON command has returned an 'FE' status to the function packet. This halt display can be indicative to the central site operator to manually dial the remote site. The TALK pushbutton on the hand-phone should be pressed to the on position while the operator dials. The TALK pushbutton remains in this position if voice contact is intended between the operator at the central site and the operator at the remote terminal. Upon termination of the conversation, the operator at the central computer site presses the DATA pushbutton on the handset and the RUN pushbutton on the 9300 console. If the remote terminal operates with automatic answering, the operator at the central site presses the DATA pushbutton on the handset and the RUN pushbutton on the control console when he hears the tone indicating that the remote terminal has switched from voice to data mode. (The TURN-ON command must be in control until the DATA pushbutton on the handset has been pressed.)

When more than one line is being used concurrently, a halt display to signal a manual dial must not be used. (A line is comprised of an input line terminal and an output line terminal.) The user should design a message printout or some discernible program loop to indicate that a manual dial is requested. The TURN-ON command remains in control even after the switched connection (dial) has been established. It is not necessary to issue an additional GET macro instruction with either a TURN-ON command (for asynchronous terminals) or a LOOK FOR SYNC command (for synchronous terminals) to input the first data message to the central computer site. The initial TURN-ON command issued should either be untimed or timed long enough to handle the possibility of conversation between operators of the remote and central sites. The alternative to leaving the initial TURN-ON command in control after the dial has been established is the issuance of a TURN-OFF command (03_{16}) with a GET macro instruction after dialing and then issuing a GET macro instruction with an appropriate TURN-ON or LOOK FOR SYNC command. In a private line situation where dialing is not needed, it is still necessary to condition the modem of the 9200/9300 System with a TURN-ON command.

6.3. INITIALIZING A MANUAL DIAL, HALF-DUPLEX TRANSMIT PROGRAM

The first command issued must be a TURN-ON command to the input line terminal of the DCS-1 or DCS-4. The dialing procedures for a manual-dial half-duplex transmit program are the same as those previously described in 6.3. The input line terminal must execute the TURN-ON command prior to issuing the first PUT (send data) macro instruction to the output line terminal. The purpose of this is to prevent a "loop back" situation in the half-duplex line procedures. The TURN-ON command must be in force until after the DATA pushbutton on the handset is pressed.

6.4. AUTOMATIC DIALING

Automatic dialing is achieved by means of a dialing adapter unit. This unit is operative on the lowest priority output line terminal of the DCS-4. The dialing adapter restricts automatic dialing to only one of four possible lines in the DSC-4.

The digits of the telephone number to be called include three groupings: an access code (usually the number 1), an area code (such as 215 for Philadelphia), and the exchange and telephone number (such as MI-6-9000). All digits in the telephone number must be represented as hexadecimal values ranging from 00 through 09. Therefore, the MI exchange shown in the example would be expressed as '0604'. The telephone number and prefix codes are defined as a hexadecimal constant, suffixed by an EOD (End of Dial) constant 'OC'. The following is an example of the coding required:

LABEL (1)	OPERATION (10)	OPERAND (16)
DIAL	DC	X'01020105060406090000000C'
*		1 2 1 5 M I 6-9 0 0 0 EOD
*		0=ZERO

The dialing adapter utilizes the 'answer signal' data set status for determining a call connection. After the dialing is completed, the automatic call unit waits for an answer to be returned from the called site. When the answer signal is detected, the auto calling unit transfers the telephone line to the data set. This indicates that the associated data set is in the data mode. The indication the central processor receives from the DA by means of the output line terminal is a good dial. Since an answer signal is expected, a TURN-ON command to the input line terminal must precede a 'dial' to the output line terminal. If, for some reason after the dial has been completed and there is not a 'data set status' signal, then a signal called 'abandon call and retry' will enter the adapter unit. This signal will indicate a bad dial to the central processor. The bad dial may be caused from a busy line, no answer, wrong number, and so forth. A unit check interrupt on the output line terminal is the status presented for a dialing error. The second sense byte (file + 83) of the File Control Table will contain a value of 10 if the dial is unsuccessful. The GCCR will decode a good dial to a 'FF' (successful completion) software status in the I/O function packet. The automatic dial procdeure by its nature of determining dial status must violate the half-duplex "loop-back" rules. The other exception to this rule is the DCS test procedure.

6.5. CONTINUOUS READ PAPER TAPE ASR'S

In many ASR 33 and 35 situations, the entire spool of paper tape will be transmitted nonstop to the 9200/9300 central site after the central site has transmitted a START READ command to the ASR. The End-of-Text character (03_{16}) in each block of data terminates the input line terminal of the 9200/9300 central site; however this does not terminate the transmission from the ASR35. The ETX character in each block of the tape is normally followed by some predetermined number of Rubout (Delete) characters to ensure that the period of "off line" time from ETX line termination to subsequent issuing of a TURN ON command to the input line terminal is sufficient to protect against data loss. This situation requires that indicator coding be used. Anytime a terminating interrupt occurs on the input line terminal, the user must tell the GCCR to reissue the TURN ON packet because the paper tape reader on the ASR is effectively in a continuous carrier mode until it senses a control "S" (stop) character or physically runs out of tape.

6.6. AUTOMATIC ANSWERING RESTRICTION

Automatic answering has a slight software restriction. The normal sequence of events for automatic answering is as follows:

(1) OPEN communications to condition the operating system to handle communications interrupts. DO NOT ISSUE A FUNCTION TO THE DCS.

(2) When a remote terminal dials the 9200/9300 central site, a unit check interrupt occurs on the input line terminal and a ring indicator is indicated in the sense bytes.

(3) A TURN ON command should then be issued to the input line terminal to clear the ring indicator and automatically answer the ring.

GCCR would have no way of fielding the unit check interrupt unless a packet was in control. The determination of a ring indicator would also be a user's responsibility. upon notification of a unit check status. Therefore, the user must issue a GET instruction with a TURN ON command without a time limit if he is expecting remote sites to dial the 9200/9300 central site. He may use indicator coding entered by a unit check status to determine if a ring indicator condition exists. If a ring indicator signal exists, the user marks the packet software status byte to reissue the packet.

UNIVAC

SPERRY RAND

UP-7816